Copie de devoirs et des examens — ورقة الفروض و الامتحانات

les champs d'informations sont obligatoires

Date ..... تاريخ

Nom et Prénom ..... الاسم و اللقب

Spécialité : BTS Réseaux et Systèmes Informatiques ..... تخصص

N° d'inscription: ..... رقم التسجيل

Module : Algorithme ..... المادة :

Devoir n° 03 ..... فرض رقم :

Cycle : 01 دورة :

Wilaya : Dr ALGER الولاية :

Exercice N°01 :

```
Program Pluriel_OU;
Fonction Determine_plur (Entrée mot : chaine de char): booléen;
Debut
    Si (mot = 'bijou') ou (mot = 'hibou') ou (mot = 'caillou')
    ou (mot = 'genou') ou (mot = 'pou') ou (mot = 'joujou')
      Alors Determine_plur := Vrai
    Sinon Determine_plur := Faux;
Fin;
var mot : chaine de char;
    Sufx : chaine de char (2);
    long : Entier;
```

```
Debut
Repeat Ecrire('Entrez un mot singulier qui se termine par
                "ou" ou tapez sur "Entrer" pour quitter');
    Lire(mot); long := length(mot);
    Si long>2 Alors suf := mot[long-1]+mot[long];
        Si suf = 'ou' Alors
            Si Determine_plur(mot) = Vrai Alors
                Ecrire('Pluriel: ', mot+'x')
            Sinon Ecrire ('Pluriel:', mot + 's'); Fsi;
        Sinon Ecrire ('Mot invalide'); Fsi;
    Sinon Si (long=1) ou (long=2) Alors
            Ecrire('Mot invalide'); Fsi;
Until mot='';
Fin.
```

Exercice N°02:

```
Program Souspile;
uses crt;
var P1, P2 : chaine de char;
    i, j, n, K, count : integer;
Debut
    Ecrire('Entrez la 1ère pile'); Lire(P1);
    Ecrire('Entrez la 2ème pile'); Lire(P2);
    count := 0; i := 1; j := length(P2);
Tant que (j <= length(P1)) Faire K := 1;
    Pour n := i à j Faire
        Si P1[n] = P2[K] alors count := count + 1;
                              K := K + 1; Fait;
```

```
        Si (count = length(P2)) Alors
            Ecrire('P2 est une sous pile de P1'); break; Fsi;
        Sinon count:=0; i:=i+1; j:=j+1; Fsi;
Fait;
Si (count <> length(P2)) Alors
    Ecrire ('P2 n'est pas une sous pile de P1'); readKey;
Fin.
```

Exercice N°03:

```
Program Polynome;
uses crt;
type lien=^term;
     term = record
          coef, exp : Entier;
          suivant, precedent : lien; Fin;
var pol1, pol2, polsom, sommet1, sommet2, sommetSom: lien;
Procedure chaineList (Entrée/Sortie pol, sommet : lien);
Debut
    Si (sommet = nil) Alors sommet := pol
    Sinon sommet^.precedent := pol; pol^.suivant := sommet;
          sommet := pol; Fsi;
Fin;
Procedure traversListAvant (Entrée/Sortie sommet : lien);
Debut
    Tant que (sommet <> nil) Faire
          Si (sommet^.suivant = nil) Alors break; Fsi;
          sommet := sommet^.suivant;
    Fait;
Fin;
```

```
Procedure LireTerme (Entrée/Sortie pol, sommet : lien);
var i : integer;
Debut i := 1;
  Repeat Ecrire ('Terme #', i); New (pol);
         Ecrire ('Entrez le coefficient :'); Lire (pol^.coef);
         Ecrire ('Entrez l'exposant :'); Lire( pol^.exp);
         ChainerList (pol, sommet); i := i + 1;
  Until (pol^.exp = 0);
  TraversList Avant (sommet);
Fin;
Debut Ecrire ('Veuillez saisir les termes du 1er polynome');
      LireTerme (pol1, sommet1);
      Ecrire ('Veuillez saisir les termes du 2eme polynome');
      LireTerme ( pol2, sommet2);
  Tantque (sommet1 <> nil) ou (sommet2 <> nil) Faire
        New (polsom);
        Si (sommet1^.exp = sommet2^.exp) Alors
          polsom^.exp := sommet1^.exp;
          polsom^.coef := sommet1^.coef + sommet2^.coef;
          ChainerList (polsom, sommetSom);
          sommet1 := sommet1^.precedent;
          sommet2 := sommet2^.precedent; Fsi;
        Sinon Si (sommet1^.exp > sommet2^.exp) Alors
          polsom^.exp := sommet1^.exp;
          polsom^.coef := sommet1^.coef;
          ChainerList (polSom, sommetSom);
          sommet1 := sommet1^.precedent; Fsi;
```

```
        Sinon polsom^.exp := sommet2^.exp;
              polsom^.coef := sommet2^.coef;
              chainerList(polsom, sommetSom);
              sommet2 := sommet2^.precedent; Fsi;
    Fait;
    traversListAvant(sommetSom);
    Ecrire('Le polynome somme est=');
    Repeat
        Si (sommetSom^.precedent=nil) Alors
            Ecrire(sommetSom^.coef, 'x', sommetSom^.exp)
        Sinon Ecrire(sommetSom^.coef, 'x', sommetSom^.exp, '+');
        Fsi; sommetSom := sommetSom^.precedent;
    Until (sommetSom=nil);
    Readkey;
Fin.
```